

Figure 1a

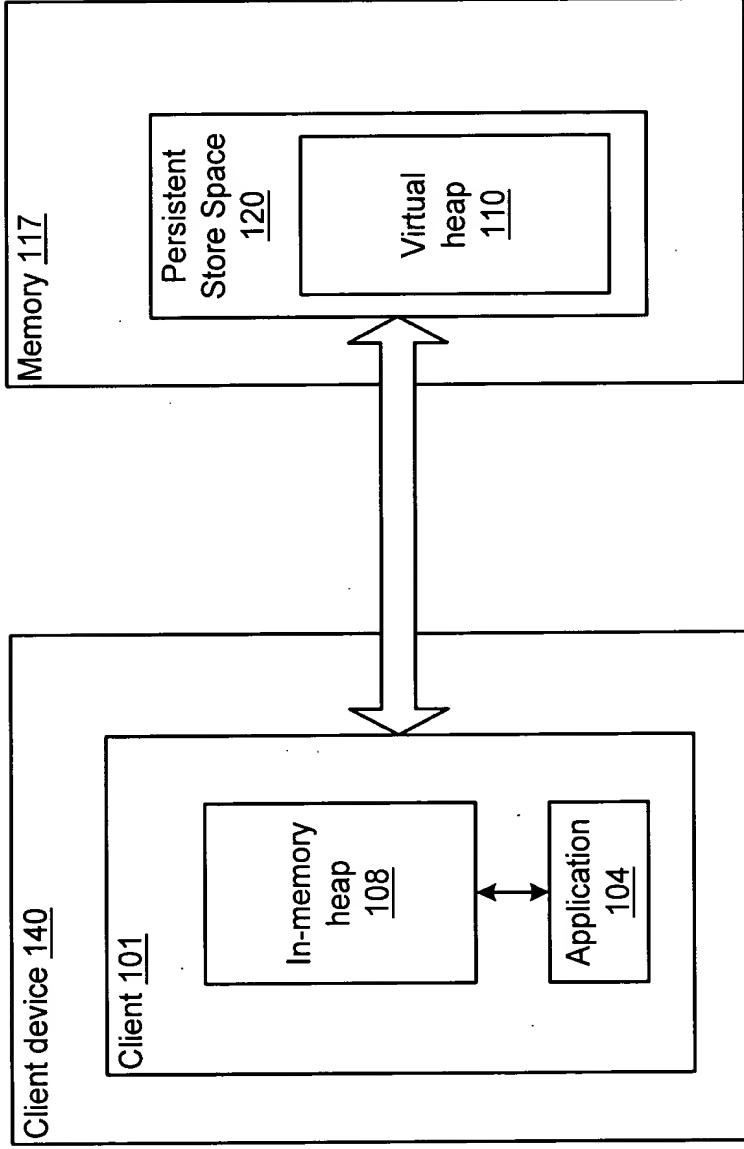


Figure 1b

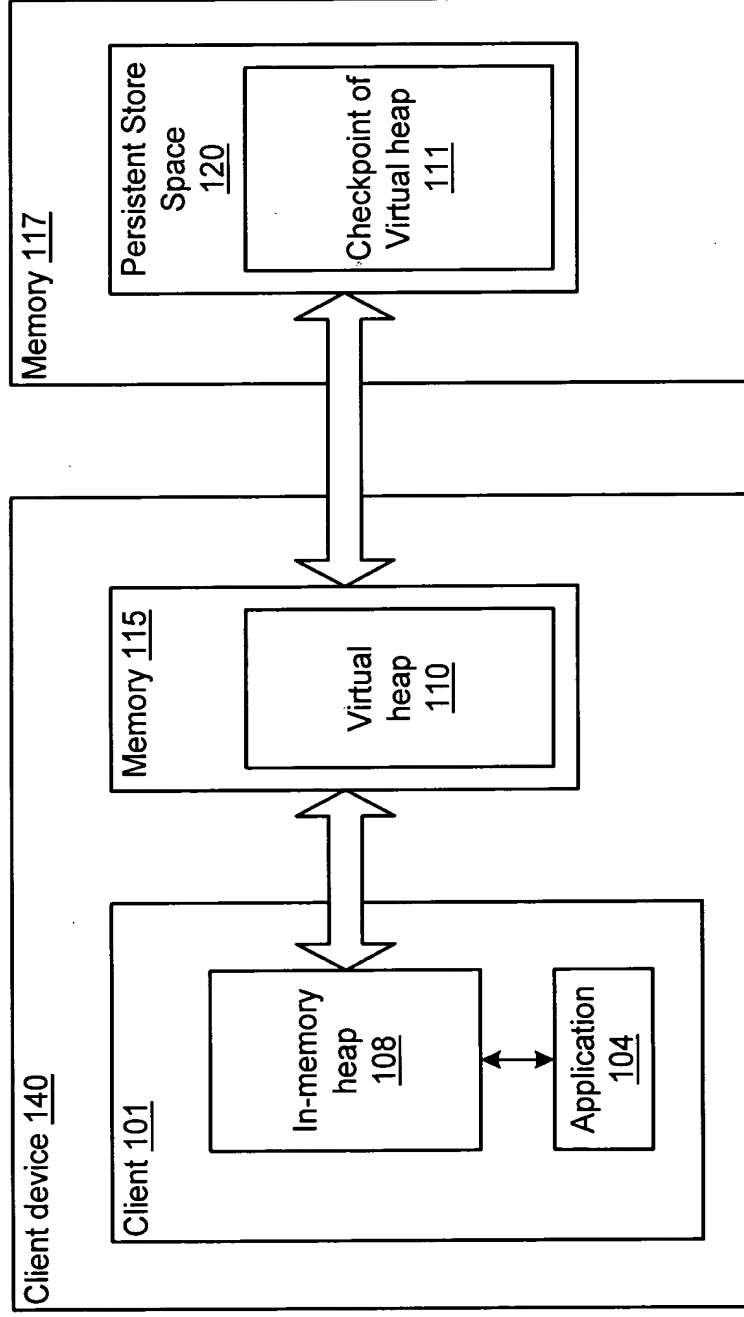


Figure 1c

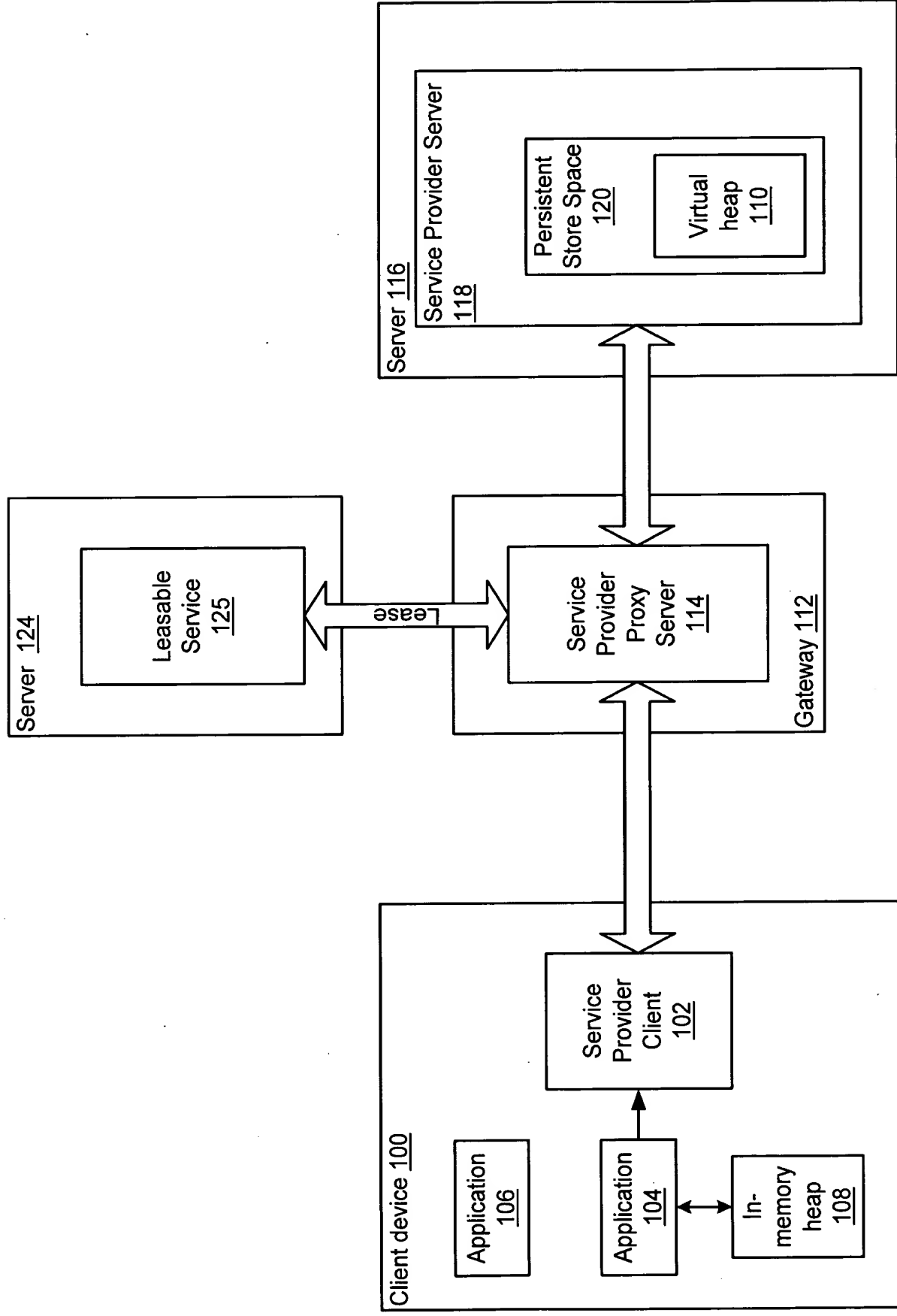


Figure 1d

```
graph TD
    subgraph VH [Virtual Heap 148]
        subgraph ACDA [Application code and data 152]
            L2S[Lease to service 164]
            L2R[Lease to resource thru system code 156]
        end
        subgraph SCDA [System code and data 154]
            NM[Native method 158]
        end
    end
    L2S --> LRS[Local or remote service 166]
    L2R --> NM
    NM --> CARS[Code for accessing system resource 160]
    subgraph NC [Native code 150]
        CARS
    end
    CARS --> SR[System resource 162]
```

Figure 1e is a block diagram illustrating a process flow. The diagram is organized into three main vertical sections. The top section, labeled "Virtual Heap 148", contains two sub-blocks: "Application code and data 152" and "System code and data 154". Within "Application code and data 152", there are two boxes: "Lease to service 164" and "Lease to resource thru system code 156". An arrow points from "Lease to service 164" to a box outside the Virtual Heap labeled "Local or remote service 166". Another arrow points from "Lease to resource thru system code 156" down to a box in the "System code and data 154" section labeled "Native method 158". From "Native method 158", an arrow points down to a box in the "Native code 150" section labeled "Code for accessing system resource 160". Finally, an arrow points from "Code for accessing system resource 160" down to a box at the bottom labeled "System resource 162".

Figure 1e

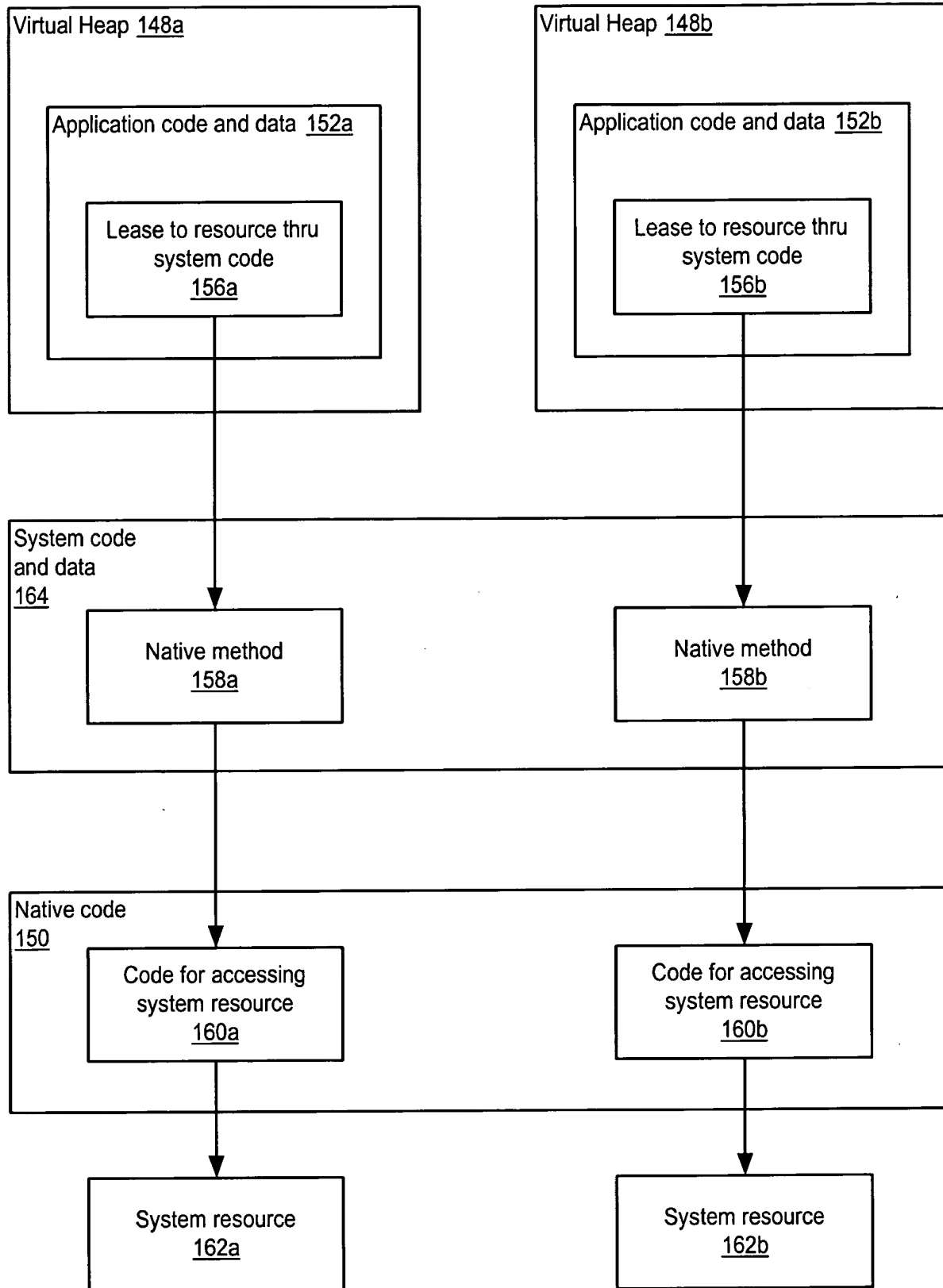


Figure 1f

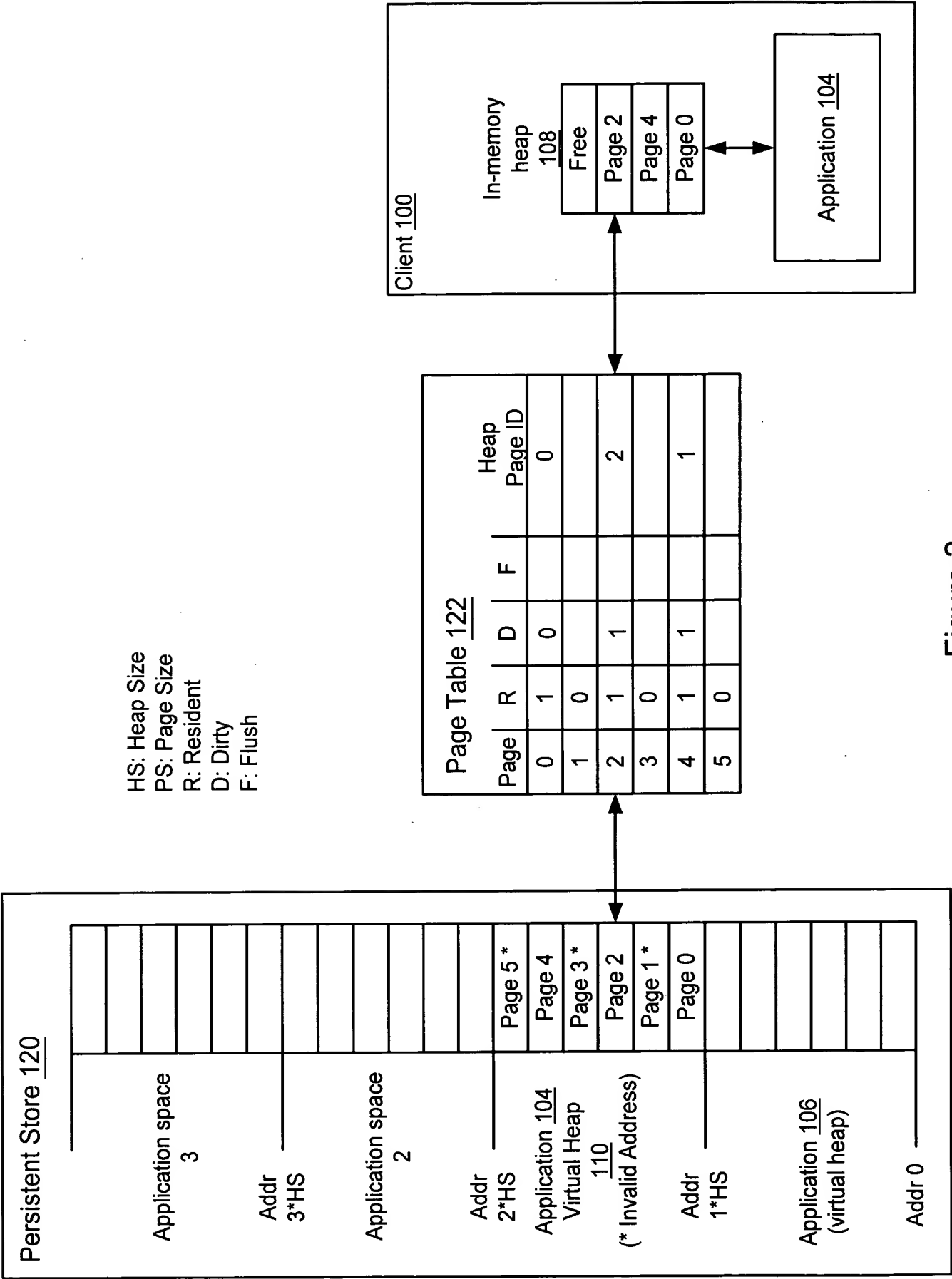


Figure 2

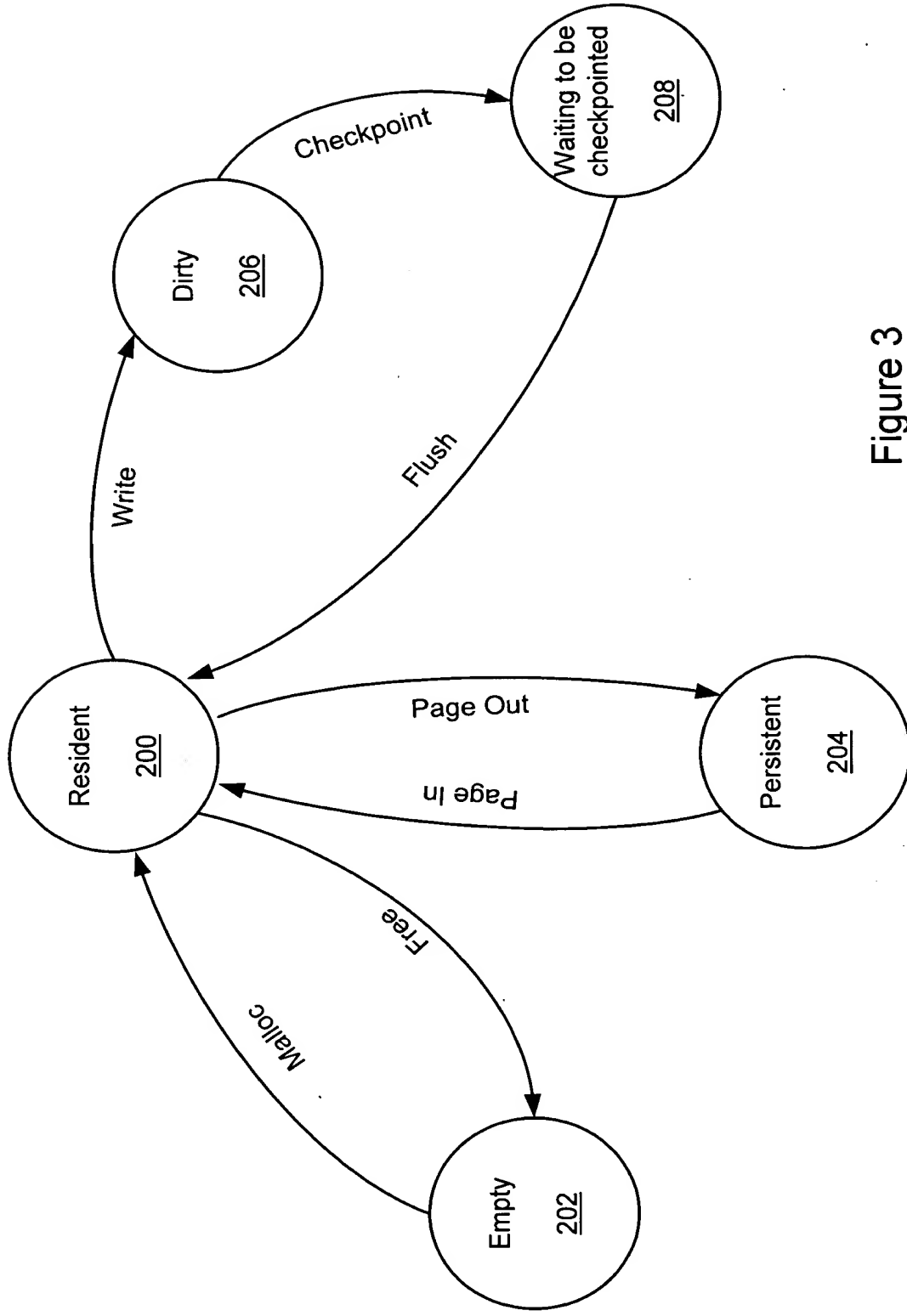


Figure 3



Determine the store page ID  
300



Determine the location of the page in the heap  
via the page table  
302



Compute the in-memory heap address  
304

### Figure 4

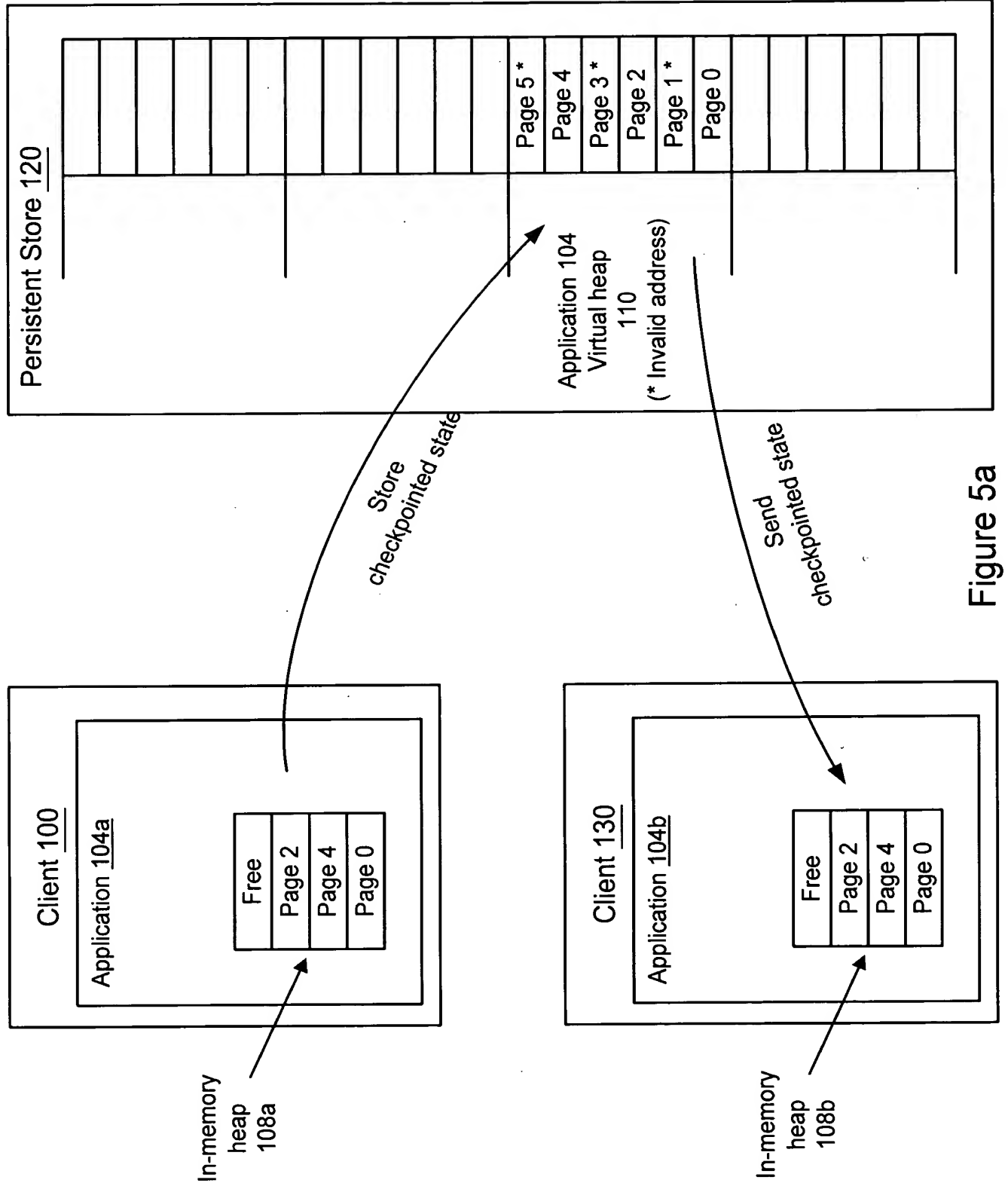


Figure 5a

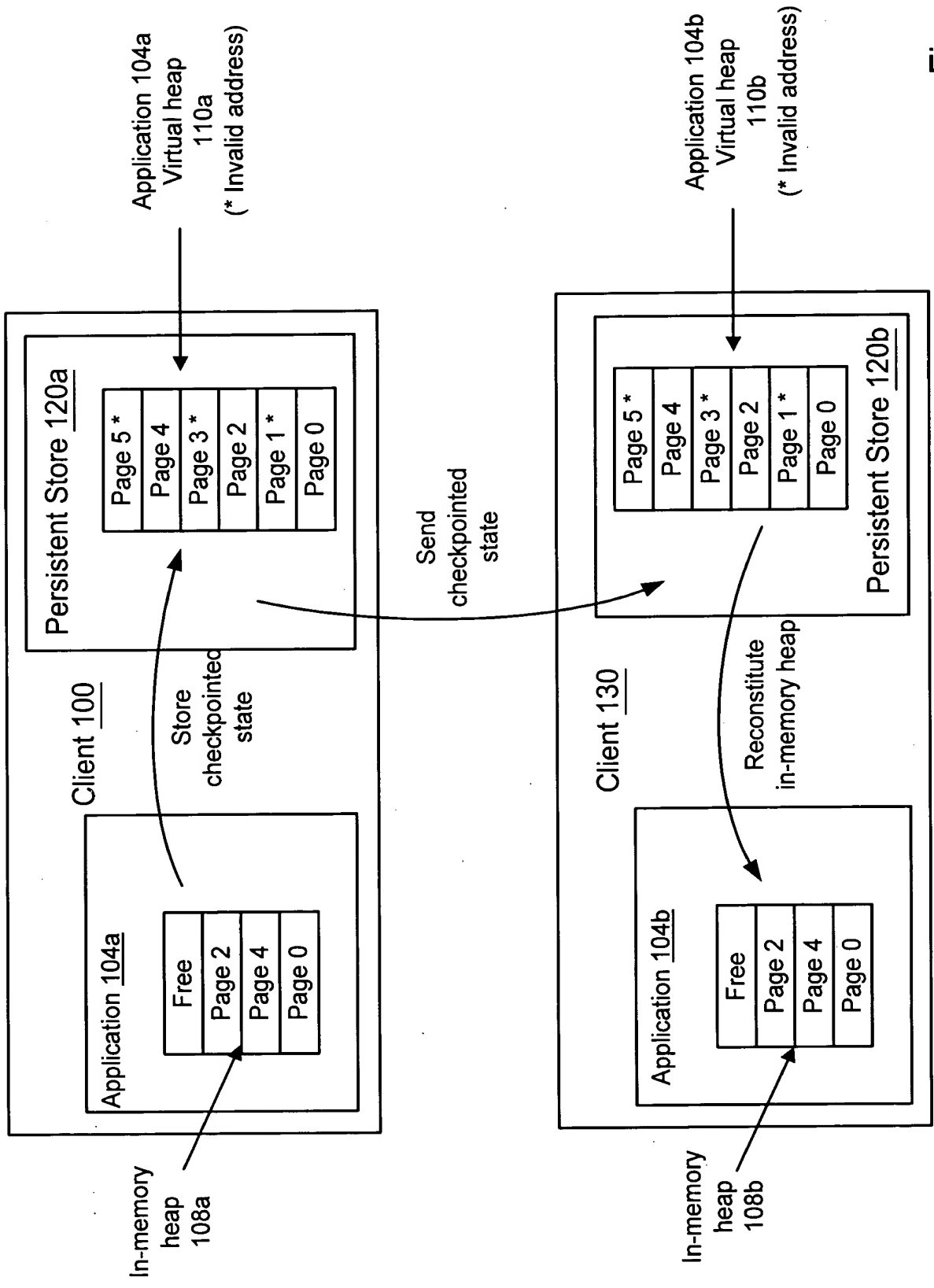


Figure 5b

```

graph TD
    320[Store the last checkpointed state of the application to the persistent store  
320] --> 322[Expire external leases to resources  
322]
    322 --> 324[Send the stored last checkpointed state of the application to the node where the application is to migrate  
324]
    324 --> 326[Receive the stored last checkpointed state of the application on the node to which the application is migrating  
326]
    326 --> 328[Commit the send transaction on both the sending and receiving nodes  
328]
    328 --> 330[Reconstitute the last checkpointed state into a new in-memory heap on the node where the application is migrating  
330]
    330 --> 332[Re-establish external leases  
332]
    332 --> 334[Resume the application on the node where it migrated  
334]

```

### Figure 6

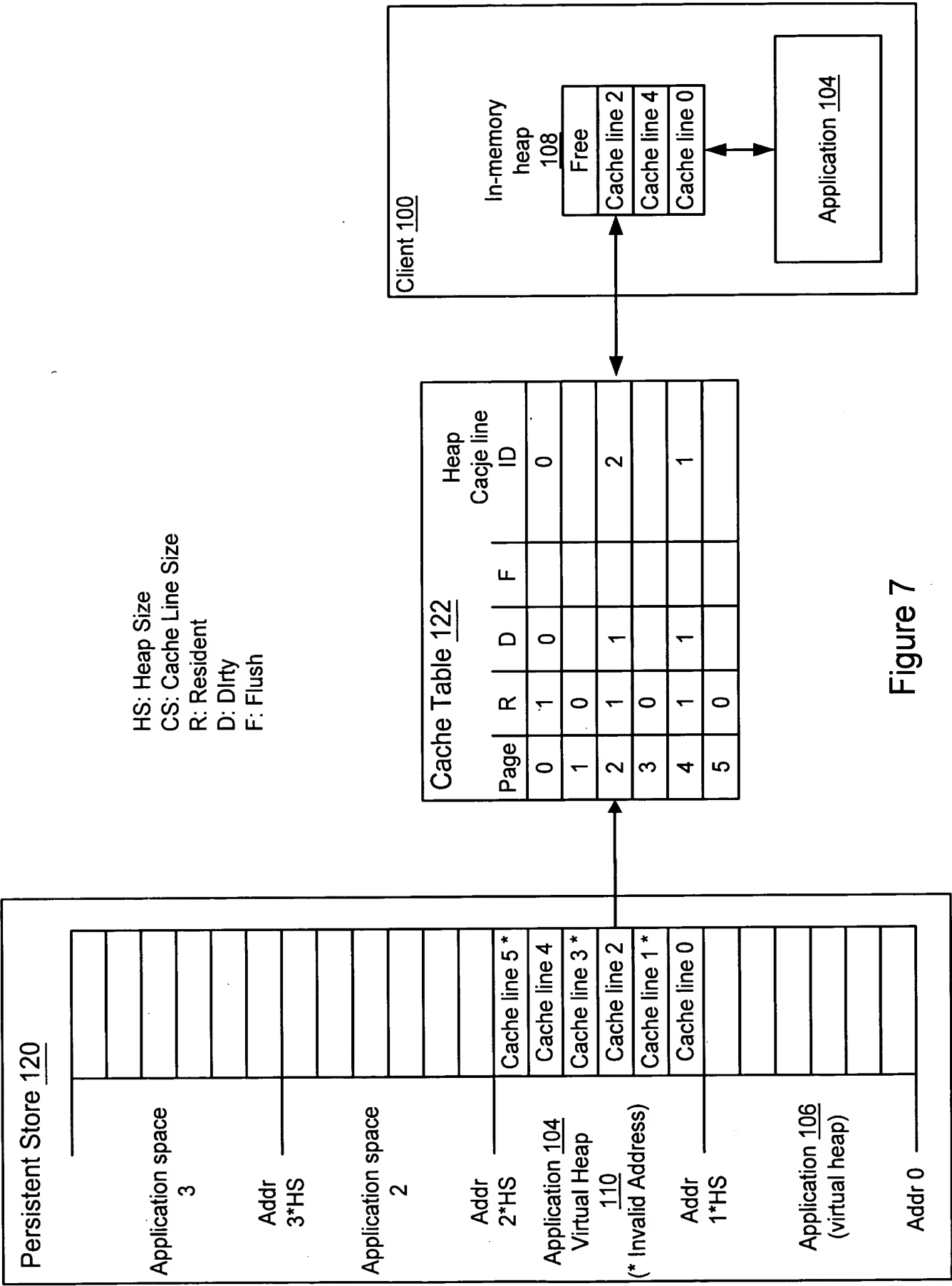


Figure 7

002090-82028560

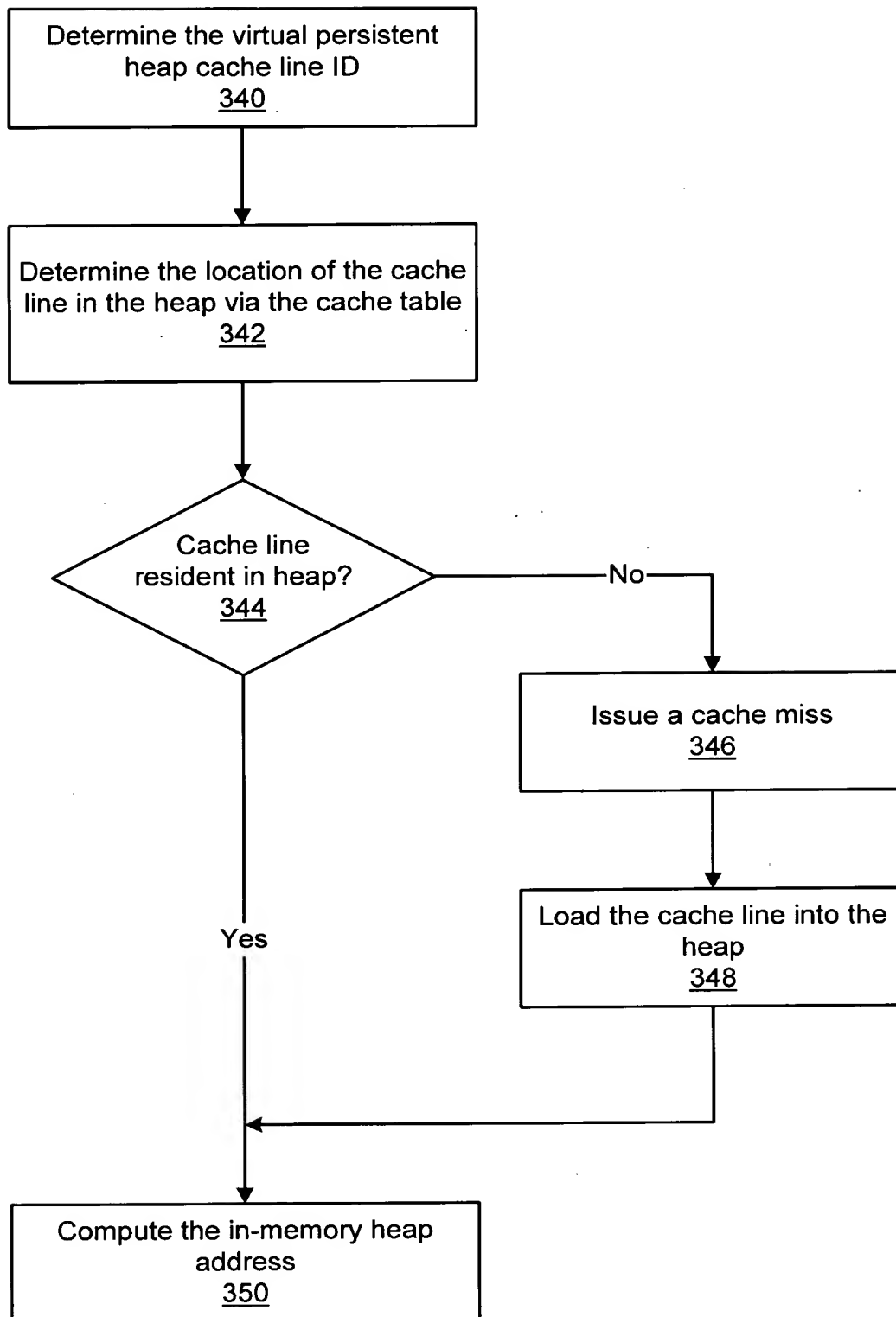


Figure 8

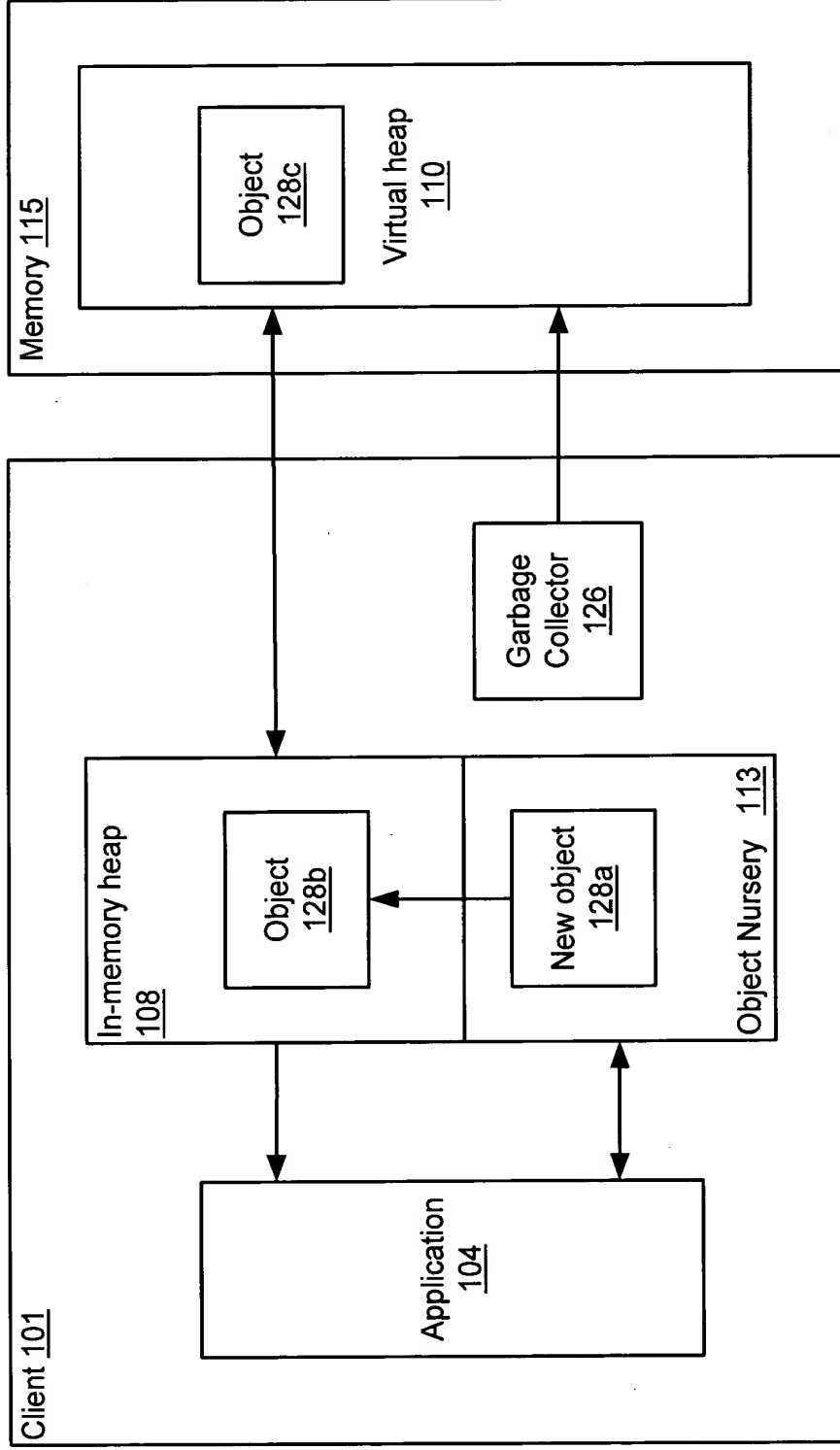


Figure 9

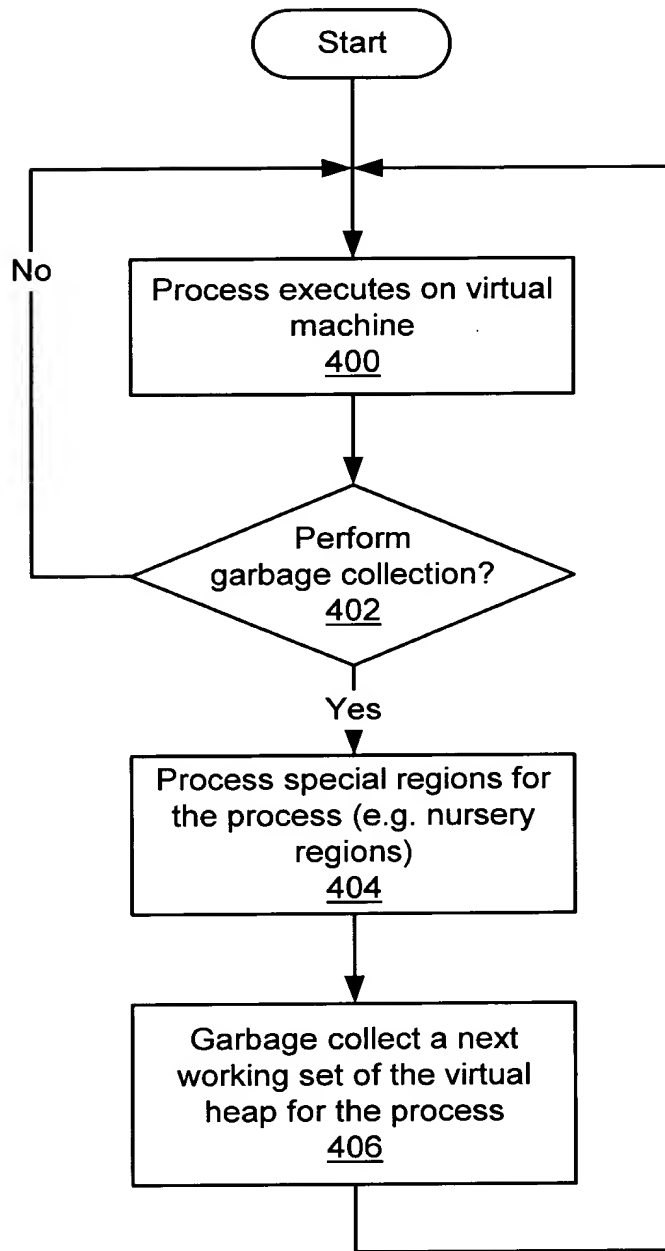


Figure 10a



```
graph TD
    Start([Start]) --> Examine[Examine object in Nursery Region  
444]
    Examine --> Referenced{Object referenced by application?  
446}
    Referenced -- Yes --> Move[Move object to in-memory heap region  
448]
    Referenced -- No --> Delete[Delete object from Nursery Region  
450]
    Move --> More{More objects?  
452}
    Delete --> More
    More -- Yes --> Examine
    More -- No --> Done([Done])
```

Figure 10b

```

graph TD
    406([Garbage collect one or more working sets of the virtual heap  
406]) --> 418[Flush in-memory heap  
418]
    418 --> 420[Examine code and data (objects) in the one or more working sets  
420]
    420 --> 422[Mark unused objects in the one or more working sets for removal  
422]
    422 --> 424[Remove marked code and data (objects) in the one or more working sets  
424]
    424 --> 426[Compact the one or more working sets  
426]
    426 --> Done([Done])

```

Figure. 10c

```

graph TD
    500[Open Store Heap  
500] --> 502[Read one or more cache  
lines from the store heap  
502]
    502 --> 504[Cache the one or more  
cache lines to the cache  
heap  
504]
    504 --> 506[Commit transaction  
506]
    506 --> 530[Close Store Heap  
530]

```

```

graph TD
    500[Open Store Heap  
500] --> 510[Read one or more cache  
lines from the cache heap  
510]
    510 --> 512[Write the one or more cache  
lines to the store heap  
512]
    512 --> 514[Commit transaction  
514]
    514 --> 530[Close Store Heap  
530]

```

```

graph TD
    500[Open Store Heap  
500] --> 520[Delete one or more cache  
lines from the store heap  
520]
    520 --> 522[Commit transaction  
522]
    522 --> 530[Close Store Heap  
530]

```

Figure 11c